Is this the future of DNS recursive resolving function?

Sara Dickinson sara@sinodun.com
sinodun.com
@SinodunCom

Jan Zorz zorz@isoc.org
Internet Society

DNS-over-TLS (DoT) Status

Date	Event
2015 - 2018	**Implementations**: Clients: Android Pie, systemd, Stu Servers: Unbound, Knot resolver, dnsd
2015 - now	Set of 20 test DoT servers
Nov 2017	Quad9 (9.9.9.9) offer DoT
Mar 2018	Cloudflare launch 1.1.1.1 with DoT
Jan 2019	Google launch 8.8.8.8 with DoT

System stub resolvers: Need native Windows & macOS/iOS support

Easy to run a DoT server

Encrypted DNS: the good… ✓

- Defeats **passive surveillance**

Opportunistic DoT: just need IP address (Android Pie default)

- Server **authentication** if a name is **manually configured** (PKIX or DANE - RFC8310)
  - Prevents redirects, can't intercept DNS queries
  - Increases 'trust' in service (DNSSEC, filtering…)

Strict DoT: need a name too

- **Data integrity of transport** - can't inject spoofed responses

Encrypted DNS: the bad & ugly…

- **SNI still leaks** (but not for long! draft-rescorla-tls-esni)
- A dedicated port (853) can be **blocked** (443 fallback)
- **Resolver** still sees all the traffic (who do you ‘trust’?)

Encrypted traffic bypasses local monitoring & security policies

- If using a resolver NOT on the local network (not available)
  - Breaks Split horizon DNS (fallback possible), leaks internal names. Similar to e.g. using 8.8.8.8 but….

How is DoH different to DoT?

Specification differences

- **A Use case (of many):** “allowing web applications to access DNS information via existing browser APIs”
- **Discovery - MUST use a URI template (not IP address)**
- **Two models:**
  - **Dedicated** connections (only DoH traffic) - hard to block
  - **Mixed** connections (send DoH on existing HTTPS connections)
    - Better privacy? Not leaking queries

No ‘Opportunistic’

Impossible to block JUST DNS traffic

New privacy concerns

- **Increased tracking**: HTTP headers allow tracking of query via e.g. ‘User-agent’ (application), language, etc.

DoH Status

“Moziflare”

	Standalone	Large Scale
Servers	• ~10 other test servers	• Cloudflare (https://cloudflare-dns.com/dns-query) • Google (https://dns.google.com/experimental) • Quad9 (https://dns*.quad9.net/dns-query) • 3 flavours of service

	Client	Servers
Implementations	• **Firefox** config option • Chrome/Bromite • Android ‘Intra’ App • Cloudflared • Stubby (next release) • Various experimental	• **dnsdist (WIP)** • **Knot resolver (patches)** • Various experimental

DNS in Browsers

DoC - DNS over Cloud !!!

- Some already have their own DNS stub (e.g. Chrome)
- Some already use encrypted DNS (Yandex, Tenta)

Dedicated DoH connections

- **Firefox had DoH since 61, not enabled by default**
- **Firefox experiment being performed….**

- Chrome has a DoH implementation (not exposed, not advertised)
  - Recent a PR to add config option
  - And Google has a handy recursive resolver service in 8.8.8.8…

Browser vendors control the client and update frequently.

DoH in Browsers

- Why encrypt directly from the browser? Browser folks say:

OS's are slow to offer new DNS features (DoT/DoH)

Selling point: "we care about the privacy of our users"

Performance: "reduce latency within browser"

- Why DoH, not DoT? Mozilla's answer:

Integration: "leverage the HTTPS ecosystem"

HTTPS everywhere: "it works… just use port 443, mix traffic"

Cool stuff: "JSON, Server Push, 'Resolverless DNS'…."

DNS 2.0?

Discussion points:

1. Centralization of DNS recursive resolving function?
2. Every app on end-host using it's own DoH recursing server?
3. Troubleshooting and support of end-hosts?
4. Privacy concerns? How would we know which DoH server each app is using?
5. Who are we sending our DNS data to? Is it easy to know? Easy to configure?
6. Who do we trust?
7. Should we go DoT or DoH way?
8. Is it fair that apps enable DoH and selection of their preferred DNS server by default?
9. Any other concerns?

Add-on: Read after-FOSDEM blog post
https://blog.powerdns.com/2019/02/07/the-big-dns-privacy-debate-at-fosdem/